# Premier League predictive learning algorithm (PLePA)

Contents

[Premier League predictive learning algorithm (PLePA) 1](#_Toc43793260)

[Glossary 2](#_Toc43793261)

[Section 1 Preparation and planning 3](#_Toc43793262)

[1.1 Project Title 3](#_Toc43793263)

[1.2 Project description 3](#_Toc43793264)

[1.2.1 What is the problem and why it is a problem 3](#_Toc43793265)

[1.2.2 Benefits of solving the problem 3](#_Toc43793266)

[1.2.3 Scope of problem 3](#_Toc43793267)

[1.2.4 Format of solution 7](#_Toc43793268)

[1.2.5 Delivery aims 7](#_Toc43793269)

[1.3 Literature review 7](#_Toc43793270)

[1.3.1 Pre TMA03 Literature 7](#_Toc43793271)

[1.3.2 TMA03 Literature 9](#_Toc43793272)

[1.4 Work done 11](#_Toc43793273)

[1.4.1 Preparation 11](#_Toc43793274)

[1.4.2 CF1 - Set up a database to store the Premier League and algorithm data 12](#_Toc43793275)

[1.4.3 CF2 – Develop the Random Forest algorithm 12](#_Toc43793276)

[1.4.4 CF3 – Develop the K-Nearest Neighbour (k-NN) algorithm 15](#_Toc43793277)

[1.4.5 Progress to date against requirements 16](#_Toc43793278)

[Section 2 Review and reflection 17](#_Toc43793279)

[2.1 Review of current stage of development 17](#_Toc43793281)

[2.1.1 Evaluation criteria used effectively 17](#_Toc43793282)

[2.1.2 What went well and why 17](#_Toc43793283)

[2.1.3 What went badly and why 17](#_Toc43793284)

[2.2 Review of project management 17](#_Toc43793285)

[2.2.1 Overview of chosen SDLC 17](#_Toc43793286)

[2.2.2 Planning issues 17](#_Toc43793287)

[2.2.3 Risk and resource management 17](#_Toc43793288)

[2.3 Review of personal development 18](#_Toc43793289)

[2.3.1 Good and bad ways of working and why 18](#_Toc43793290)

[2.3.2 Skills to complete work 18](#_Toc43793291)

[References 20](#_Toc43793292)

[Bibliography 20](#_Toc43793293)

[Appendices 22](#_Toc43793294)

[Appendix 1 – Snippet of project journal 22](#_Toc43793295)

[Appendix 2 – LEPSI Review 23](#_Toc43793296)

[Appendix 3 – Table creation script 24](#_Toc43793297)

[Appendix 4 - Main function for database insert 24](#_Toc43793298)

[Appendix 5 – First draft of schedule 25](#_Toc43793299)

[Appendix 6 – Latest draft of schedule 26](#_Toc43793300)

[Appendix 7 – Resource list 27](#_Toc43793301)

[Appendix 8 – Risk list 28](#_Toc43793302)

[Appendix 9 – Skill list 29](#_Toc43793303)

|  |  |  |  |
| --- | --- | --- | --- |
| Appendix | Page | Name | Description |
| 1 | 22 | Snippet of project journal | A small portion of the project journal from 17th May to 12th June. |
| 2 | 23 | LEPSI Review | A table to show research done for LEPSI review. |
| 3 | 24 | Table creation script | The table creation script written in MySQL. |
| 4 | 24 | Main function for database insert | The main() Python function for the database\_insert module. |
| 5 | 25 | First draft of schedule | The first draft of PLePA schedule. |
| 6 | 26 | Latest draft of schedule | The most up to date draft of PLePA schedule. |
| 7 | 27 | Resource list | A table to show resources required. |
| 8 | 28 | Risk list | A table to show risks identified and how they will be managed. |
| 9 | 29 | Skill list | A table to show skill required. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Figure | Page | Name | | Description |
| 1 | 11 | Structured-case lifecycle | | An image to show the process of the structured-case lifecycle |
| 2 | 12 | Database conceptual model | | An image to show the PLePA conceptual model |
| 3 | 13 | Code snippet of Random Forest algorithm | | A small portion of the code to show how each tree is created in the Random Forest algorithm. |
| 4 | 14 | Thresholds of Random Forest algorithm | | A graph to show the thresholds of the Random Forest algorithm. |
| 5 | 14 | Correct prediction rate of training data for the Random Forest algorithm | | A graph to show the fluctuation of the prediction rate when training the Random Forest algorithm. |
| 6 | 15 | Code snippet of k-NN | | A small portion of code to show the code for the main three functions for the k-NN algorithm |
| 7 | 16 | Correct prediction rate for the k-NN algorithm | | A graph to show the prediction rate for each neighbour of the k-NN algorithm. |
| Table | Page | Name | Description | |
| 1 | 5/6 | Project scope | A table to describe each part of the project scope and a current evaluation. | |
| 2 | 9/10 | Literature review | A table review of the literature read between TMA02 and TMA03. | |
| 3 | 16 | Activity evaluation | A table to show the evaluation of the activities currently, giving detail on where each activity is up to. | |

# Glossary

PLePA – Premier League predictive learning algorithm, the name of the project.  
Premier League – The highest league in English football.  
Structured-case lifecycle – The lifecycle chosen for PLePA, it consists of multiple Conceptual Frameworks, described below.  
Conceptual Framework, also known as CF – A conceptual framework is a chunk of the lifecycle, it has a plan, process, analysis and reflection piece.   
LEPSI – Legal, Ethical, Professional and Social issues.  
Random Forest – Random Forest is one of the algorithms developed.  
k-NN – K-nearest neighbours is one of the algorithms developed.  
SDLC – Software Development Life Cycle, it is the process from beginning to end of a software project, there are many different life cycle choices.

# Section 1 Preparation and planning

## Project Title

Premier League predictive learning algorithm (PLePA)

## Project description

The project is to create a predictive algorithm for Premier League results. It will use historic data; it will use data since the 2014-15 season. It is now a trivial task to add more data if it is deemed needed because the work has been done on creating the Python scripts to insert the data and notes have been made on cleansing the data, which is a couple of simple steps. The difference between this project and similar projects out there is that all the algorithms developed will use the distance travelled for the away team as a key factor.

The result is aimed at Premier League fans and people interested in the prediction of football results. It’s also something I have an interest in solving, I do a bit of sports betting so this could prove useful for this.

The aim is to produce four separate algorithms and determine which has the most accurate results. The four algorithms I will use are Support Vector Machines (SVM), Random Forest, Naïve Bayes and k-NN. The goal is to have around a 75-80% pass rate, this will be a benefit to all football fans, fans who place bets and even possibly teams, that may be a bit far though.

If the project is unable to give good results after many iterations for each algorithm and tweaking the numbers, then it will serve as research for people who take on a similar project. There is no real issue if it not developed because it is more of a personal project and I am the only stakeholder.

There are different ways to achieve the results, the aim is to develop four separate predictive algorithms, this will give the opportunity to move on if one is not going well.

The stats data will be stored in a MySQL database and all the algorithm code will be written in Python. All code and documents will be stored in a GIT repository. A stretch goal would be to have an interface for user interaction or to pull the latest set of fixtures from a football website.

### 1.2.1 What is the problem and why it is a problem

The problem is the inability to predict football results with a good degree of accuracy. This makes things such as betting on or even just predicting premier league football results difficult. It is also interesting, for football fans and statistic fanatics, seeing the likelihood of certain teams winning in unlikely situations.

### 1.2.2 Benefits of solving the problem

The benefits of solving the problem is only that it can be used as a tool for loosely predicting premier league results. It is mainly a personal project so if it remains unsolved there is no negative impact on the wider community.

1.2.3 Scope of problem  
PLePA will have four algorithms developed and each algorithm will have a pass percentage rate, possibly with some more statistics. This will depend on what is gatherable. The algorithms will use Premier League data stored in a MySQL database. There are two stretch goals – developed a feed from a sports website to pull in latest fixtures and develop a user interface so PLePA can be distributed beyond personal use. The four algorithms will also be compared.

|  |  |  |  |
| --- | --- | --- | --- |
| Scope | Aspect of problem | Associated tasks | In or out  If out, why out? |
| Preparation stage, plan and make decisions on how the project will be carried out. | The goals give a clear indication of what the project is aiming for. This is useful for keeping the project on schedule. | Define the goals and contents of my project.  Plan the database structure to use.  Look into Python modules that will be useful.  Set up base Python project with GIT version control.  Install database software.  Find the best source for the Premier League statistics required.  Review legal, ethical, social and professional issues and decide if there are additional actions required. | In and completed. |
| Research additional material that will be useful. | Carry out research so PLePA can make informed decisions. | Research SDLC choices and decide on one for the project.  Research difference between Oracle and MySQL and decide on one of them to use.  Research similar studies for ideas and document key findings.  Research predictive algorithms. | In and completed. |
| Conceptual Framework (CF) 1 - Set up a database to store the Premier League and algorithm data. | All algorithms will need the data to work, so it is essential work to solve the problem. | Gather required Premier League data.  Cleanse the data gathered.  Create the database.  Create a database reset script. | In and completed. |
| CF2 - Develop the Random Forest algorithm. | The first of the four algorithms to be developed for PLePA. | Research algorithm.  Plan how algorithm will be coded.  Code algorithm.  Test algorithm.  Evaluate the algorithm.  Produce statistics on algorithm. | In and completed. |
| CF3 - Develop the k-NN algorithm. | The second of the four algorithms to be developed for PLePA. | As above. | In and completed. |
| CF4 - Develop the Support Vector Machines, SVM, algorithm. | The third of the four algorithms to be developed for PLePA. | As above. | In, not yet started. |
| CF5 - Develop Naïve Bayes algorithm. | The fourth of the four algorithms to be developed for PLePA. | As above. | In, not yet started. |
| CF6 - Develop a feed from a sports website to pull in the latest fixtures. | This could allow automation of predicting upcoming results. | Research what websites could be used to pull in the Premier League data required.  Development for the data to come in from the chosen feed. | Most likely out, but not confirmed. Based on how long the first two algorithms have taken to develop, it is unlikely there will be scope to include this item because it is only a nice to have. |
| CF7 - Develop a user interface so PLePA can be distributed. | This would allow other users to use, this is not essential since it is a personal project and there are no additional stakeholders. | Sketch a user interface for screen/s to use the PLePA algorithms developed.  Create the screen/s which have been sketched. | Most likely out, but not confirmed. Based on how long the first two algorithms have taken to develop, it is unlikely there will be scope to include this item because it is only a nice to have. |

Table 1, Project scope.

1.2.4 Format of solution  
The solution will be a program which can quickly predict results for multiple games for each algorithm. Based on previous predictions, the algorithm will also be able to give a likelihood of the predictions being correct.

### 1.2.5 Delivery aims

PLePA will have four main deliverables: Premier League stats stored in a database, a working predictive algorithm to predict the results, an interface for user interaction and a feed from a football website. Each deliverable is discussed below.

* Premier League stats stored into the database must have been complete with a years’ worth of data, ideally it will have three or more years’ worth have data but a best-case scenario, time permitting, there will be five years of Premier League stats stored.
* There will be at least two predictive algorithms developed, this is a must have for PLePA to allow for comparisons between the algorithms. It should be within scope to achieve three, but, also, without any development obstacles, four is easily achievable.
* There are two nice to haves if time goes well, firstly, an interface for user interaction will be developed, there are two parts to this, the first is for the user to load up an executable, select two clubs and they will receive a prediction from the algorithm. The second would allow the user to select the algorithm to predict with.
* The second nice to have is a feed from a football website, the upcoming fixtures would be pulled from a website, the user would select the algorithm and the predicting results would be displayed.

## Literature review

### 1.3.1 Pre TMA03 Literature

Each external material used will be discussed below:

* TM351 part 1 -26 (The Open University, 2014) is the Data management and analysis module provided by the OU, this was pivotal to PLePA because it was the inspiration for the project, and it provided the skills for developing and managing a database and algorithms.
* Machine Learning Algorithms in Python (Dataflair Team, September 2018) was a useful read, it was not too old and gave a good account of developing algorithms in Python.
* Choosing a Lifecycle Model (The Open University, 2014) gave insight into multiple lifecycles that were available and discussed how to go about choosing one for the project.
* SDLC Models Explained: Agile, Waterfall, V-Shaped, Iterative, Spiral (Osetskyi, 2017) explained in detailed some of the popular SDLC choices and what they are best for.
* Structured-case: A methodological framework for building theory in information systems (J.M. Carroll et al., 2000) provided a good insight into a SDLC which was unknown to the author of PLePA, this went on to be the SDLC used for PLePA.
* Data-files: England (Football-Data,2020) is where the data used for match history stats was collected from.
* 2019-2020 Premier League Stats (FBREF, 2020) is where the stats used for league positions was collected from, they also store historical data, so the last five seasons’ worth of data was also from that website.
* The Beautiful Game: Predicting the Premier League with a random model (Nguyen, 2018) and Betting on the English Premier League (Campanelli, 2019) was research reading to see other views on solving the same problem. It was helpful to see the different approaches, what went well, and problems that were encountered.
* Legal, Social, Ethical and Professional issues (The Open University, 2012), The Data Protection Act (UK Government, 2018), Equality Act 2010 (UK Government, 2015), Computer Misuse Act 1990 (UK Government, 1990), How copyright protects your work (UK Government, n.d), What is the freedom of Information Act? (ICO, n.d) and Code of Conduct for BCS Members (BCS, 2019) were all important to read to ensure PLePA was not in breach of any Legal, Social, Ethical or Professional issues. If PLePA was in breach, these articles would have helped to resolve and steer the project in the correct way. They also helped to keep the author up to date on the government’s latest laws and guidance.
* An Implementation and Explanation of the Random Forest in Python (Koehrsen, 2018), How Random Forest Algorithm Works in Machine Learning (Synced, 2017) and How the Random Forest Works in Machine Learning (Polamuri, 2017) were all research done in preparation for development of the first algorithm, Random Forest. The articles all explained how they had developed the Random Forest algorithm in Python, which is the same development language used for PLePA, so this was suitable research for the project.

### 1.3.2 TMA03 Literature

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Literature | Presentation | Relevance | Objectivity | Method | Provenance | Timeliness |
| How to Implement Random Forest From Scratch in Python (Brownlee, August 2019) | Very good, each code snippet had a good explanation to what it was doing. | Very relevant, this was the algorithm being developed in the chosen scripting language. | It is the author’s views on implementing the algorithm, there will always be multiple ways to implement the algorithm. | The author has explained his method and given detailed explanation on why. | The author has lots of articles on developing algorithms in Python with good reviews. | The material was updated less than a year ago. |
| How to Implement The Decision Tree from Scratch in Python (Brownlee, December 2019) | As above. | The research carried out in How to Implement Random Forest From Scratch in Python (Brownlee, August 2019) triggered this research, it was useful to understand the development of each tree within the Random Forest. | As above. | As above. | As above. | As above. |
| Develop k-Nearest Neighbors in Python from Scratch (Brownlee, 2019) | As above. | Very relevant, this was the algorithm being developed in the chosen scripting language. | As above. | As above. | As above. | As above. |
| Implementing a Random Forest Classification Model in Python (Huneycutt, May 2018) | The presentation was clear, but the information provided was not. | Not that useful, a lot of assumed knowledge and explanations for examples not given clearly. | As above. | Not clearly explained why the author has chosen to do things in certain ways. | The author has lots of articles on Python projects. | The material was released in May 2018 with no updates, so it is not too outdated. |
| In-Depth: Decision Trees and Random Forest (VanderPlas, March 2016) | The presentation was clear, everything followed logically. | It was mostly useful, however, some of the information given was out of the scope of PLePA. | As above. | Mostly explained well, the author has assumed some knowledge. | It’s an article from the Python Data Science Handbook which is a well-known book. | It’s a few years old so some of the material may be outdated. |
| K Nearest Neighbor Algorithm In Python (Maklin, July 2019) | Very basic but a clear layout. | This was a good starting point to research k-NN, it was quite basic and didn’t go into too much depth. | As above. | There is not a lot of information, but it is the author’s own implementation. | It is posted on towardsdatascience.com so assumed to be mostly reliable. | Less than a year old so unlikely to be too outdated if at all. |
| KNN Classification using Scikit-learn (Navlani, August 2018) | Great layout, pictures and code snippets used to break up the text and help explain. | PLePA did not end up using Scikit-learn but the article was still useful for Python research on k-NN. | As above. | All information provided is explained clearly. | It is posted on DataCamp’s website so is expected to be reliable. | Under two years old so will be mostly in date still. |
| 1.6 Nearest Neighbors (Buitinck et al., n.d) | Lots of text which makes it hard to keep focus, but the occasional image did help to explain some parts. | This was mostly useful; some was outside the scope of PLePA. PLePA did not end up using the module but the information was still useful | It is the implementation of the algorithm using the specific Scikit-learn Python module. | The author developed the module so will be factual on how to use it. | It is a widely used Python module so it can be trusted. | No date for this specific article but it is expected the module will be kept up to date. |

Table 2, Literature review

## Work done

All work done followed the schedule created, the latest version can be seen in Appendix 6. All work done was also documented in the project journal, a snippet of this is in Appendix 1.

### 1.4.1 Preparation

Before any development could begin, the preparation stage had to be complete. The first task to be completed was to set up a GitHub repository for storing all the project work, this ranged from the Python files to the research documents. It was important to keep everything version controlled to ensure no work was lost when overwritten, this allows for things such as the schedule to be monitored throughout the project.   
Once this was setup, the next, important, step was to choose a software development life cycle (SDLC), after doing research on multiple lifecycles (The Open University, 2014)(Osetskyi, 2017) (J.M. Carroll et al., 2000), the decided SDLC was the structured-case lifecycle. This involves multiple conceptual frameworks (CF) which each have four parts, plan, do the work, analyse and reflect. This can be seen more clearly in Figure 2 below.

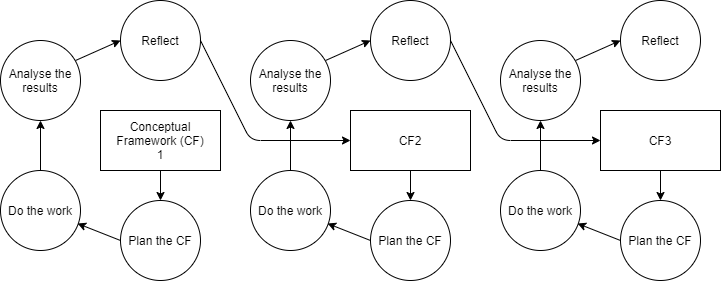


Figure 1 (Blagg, 2020) Structured-case lifecycle

Once the lifecycle was chosen, the first draft of the schedule could be created, the latest schedule can be seen in Appendix 6.   
After this, the research begun, the first part of research was to look at similar projects (Nguyen, 2018) (Campanelli, 2019), from these projects, it was useful to see things that worked well and did not work so well. One thing that was noticed is that both projects researched only used one algorithm for result prediction so this is one way that PLePA would differ from these. The research into predictive algorithms also began at this point (Dataflair Team, September 2018), it was useful to see ways of implementing the algorithms.   
Next came the search for Premier League data that could be used for PLePA (FBREF, 2020)(Football-Data,2020). The data chosen was from the seasons between 2014 and current day. This data was downloaded and cleansed, it was not a difficult task to find the data to use, the first websites checked had the data required downloadable in csv format. The cleansing of data was mainly to match up the results data with the historic league positions. There were some small differences such as Leicester in the results data and Leicester City in the league position data.   
At this point, the legal, ethical, professional and social issues (LEPSI) surrounding PLePA could be researched, investigated and determined if any work needed to be altered. This research can be seen in Appendix 2, the result of this research was that there was no work to be changed for PLePA, but the awareness of the acts and guidelines was important.

### 1.4.2 CF1 - Set up a database to store the Premier League and algorithm data

The planning stage of the CF was to think about the database structure that could be used and what data would be required for the database.  
The first of doing the work for CF1 was to create a conceptual model, the final structure decided upon for the main database can be seen below in Figure 1. This went through multiple iterations and there were a few points discussed with my tutor such as dropping pks on stadium\_name since it was not required with stadium\_id\_pk already there and using home\_team\_id\_pk\_fk and away\_team\_id\_pk\_fk instead of team\_one\_id\_pk\_fk and team\_two\_id\_pk\_fk.

A screenshot of a cell phone

Description automatically generated

Figure 2 (Blagg, 2020) Database conceptual model

Once the format had been chosen, the tables need to be created in the database, the table\_creation script can be seen in Appendix 3. After that, the stadium data was manually looked up in Google, coordinates were identified, and this data was added into a csv file. Next, the data found in the preparation stage and the stadium data created could be inserted. This was done by using a Python script which iterated over the CSV files which has been downloaded and cleansed. The script iterates over each row, inserting the correct fields into the database from the data given. The main function for this script can be seen in Appendix 4.  
Once all the data was inserted, a database reset script was created. This was useful when mistakes were made, such as data becoming invalidated because it allowed for a safe restart point where the state of the database was known.  
There was not a lot of analysis done on CF1 other than a further review of the database and data used to ensure it would all work correctly.  
CF1 went successfully without any major issues, it was hard to establish any problems because the database was not yet in use by any of the algorithms.

### 1.4.3 CF2 – Develop the Random Forest algorithm

CF2 began with researching more into the Random Forest algorithm and how it could be implemented in Python (Koehrsen, 2018) (Synced, 2017) (Polamuri, 2017) (Brownlee, Aug 2019) (Huneycutt, 2018) (VanderPlas n.d). A Random Forest algorithm is made up of multiple decision trees so further research was carried out on understanding decision trees more clearly as well (Brownlee, Dec 2019). Once the research had been complete, it was important to plan how the algorithm would work. It was decided that a new table would be created in the database to store moving thresholds for determining whether a result was a win, draw or loss for the home team. Each tree would have a score ranked between zero and three, the score for the tree would be obtained through three branches giving a score between zero and one hence the total score for the tree being up to zero. The thresholds would be trained using the data from 2014 to the end of the 2018/2019 season. This would allow the testing of the algorithm to be on the current season’s data.  
Once the development for CF2 had started, the need for a new table was identified, random\_forest\_results table, amongst other columns, this would be used to store which three functions were used, what the predicted result was and the actual result. This was needed to keep store the results from the algorithm.  
Figure 3 below shows part of the code for the random\_forest\_algorithm, “for i in range(100):” means 100 tree would be used for the forest, this was the number decided upon after testing with various numbers. “for i in range(3):” is the number of branches used for each tree, what this means is that it will go into that loop three teams and randomly select one of the five functions. The criteria for each of those five functions would determine how successful the algorithm would be.

A screenshot of a cell phone

Description automatically generated

Figure 3 (Blagg, 2020) Code snippet of Random Forest algorithm

Before the algorithm could start training, the thresholds had to be inserted into the database, the algorithm was run a few times to get a feel for what results would be produced. The algorithms then got set at 1.8 for a win and 1.2 for a draw, anything below 1.2 would be a loss. The movement of the thresholds can be seen below in Figure 4.

Figure 4 (Blagg, 2020) Thresholds of Random Forest algorithm

Each number on the x-axis represents a whole years’ worth of data used to train the thresholds, there were five years of training data and each year was used a thousand times. From Figure 4, it is clear the thresholds did not move much, the most noticeable changes were between 1000 and 2000 where the win threshold dropped, this was the 2015/16 season and between 3000 and 4000 where the draw threshold increased, this was the 2017/18 season. The correct prediction rates were also recorded during this training and ca be seen below in Figure 5.

Figure 5 (Blagg, 2020) Correct prediction rate of training data for the Random Forest algorithm

After all the training, the algorithm could run over the current seasons’ data, this gave a correct prediction rate of 46.40%. This did not go as well as expected but without other algorithms to compare it to, it was difficult to know whether this was the work of the algorithm or that the data used, mainly, the distance travelled did not have a huge impact on the result of the game. It is important to note that there are so many factors influencing a game of football which is why it is so difficult to predict.

### 1.4.4 CF3 – Develop the K-Nearest Neighbour (k-NN) algorithm

Again, CF3 started similar to CF2 with research into the algorithm (Maklin, 2019)(Buitinck, n.d)(Brownlee, Oct 2019)(Navlani, 2018). Prior knowledge to the k-NN algorithm was also obtained through OU studies in TM351 (The Open University, 2014), this meant less preparation was required for working on this algorithm. The algorithm involved using the main data and test data all in the same program, unlike the Random Forest algorithm where training the data could be seen separately to the main algorithm. The main three functions written for k-NN can be see below in Figure 6.

A screenshot of a cell phone

Description automatically generated

Figure 6 (Blagg, 2020) Code snippet of k-NN

The first function, euclidean\_distance, takes in the coordinates of two points, neighbors, and returns the distance calculated.   
The second function, get\_k\_nearest\_neighbors, takes in the training\_data, the row being classified, new\_row, and the amount of neighbours used to determine the classification, neighbor\_num, this function will calculate the distances to all the training\_data and return the closest amount of neighbours which is determined by neighbor\_num.  
The third function, get\_classification, takes in the same parameters, finds the classification for each neightbour and determines the classification for the new point by using the maximum classifications. For example, if there are ten nearest neighbours and six of them are wins, the classification for the new point will be a win.  
The results for the k-NN algorithm can be seen in Figure 7 below, the graph shows that the peak for correct prediction rate was when k was at 3. The percentage was 49.64% which is still low but is slightly better than the Random Forest algorithm at 46.40%.

Figure 7 (Blagg, 2020) Correct prediction rate for the k-NN algorithm

### 1.4.5 Progress to date against requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Criteria | Evaluation | Progress to date |
| Premier league stats in the database. | Must have. | There will be a year’s worth of data in the database. | As discussed in section 1.4.2, this activity has been completed with five years of data. |
| Should have. | Three years. |
| Nice to have. | Five years. |
| A working predictive algorithm to predict the results. | Must have. | Two algorithms will have been developed and can be used. | As outlined in section 1.4.3 and 1.4.4, this activity is currently at the must have, this will hopefully have all four algorithms by the end of the project. |
| Should have. | Three algorithms. |
| Nice to have. | Four algorithms. |
| An interface for user interaction. | Nice to have. | A user can load up an executable and select two clubs, they will then receive a prediction. | No progress. |
| Nice to have. | Further development could mean the user could select the algorithm to predict with and the results using the teams and the algorithm are displayed. | No progress. |
| A feed from a football website. | Nice to have. | The upcoming fixtures are pulled from a website and predicted using the chosen algorithm. | No progress. |

Table 3, Activity evaluation

# Section 2 Review and reflection



## Review of current stage of development

### 2.1.1 Evaluation criteria used effectively

The evaluation criteria, in Table 3, has been useful for keeping on track with the project, it is also helpful to have a glance at the table and quickly see where PLePA is currently up to. It is unlikely all activities will be completed due to the time taken to develop each algorithm. Although, the activity is less than half complete, the bulk of the work is in the second activity, having four algorithms developed. This activity is halfway complete.

### 2.1.2 What went well and why

Keeping strict on the schedule has worked well for this project, this is due to knowing there are set deadlines for each task in the schedule. Falling behind in certain tasks will push everything else back and the schedule will then become outdated.  
The development of the algorithms has been enjoyable; this is because I enjoy learning new Python code and thinking about how I can implement it in my project. It was satisfying to work on the algorithms and see results produced at the end of it.

### 2.1.3 What went badly and why

The algorithm results are a little disappointing, this is not necessarily going badly but the data chosen to use in the algorithms is not working as well as expected. It was always a gamble to use distance travelled for the away team as a factor since this is a largely untested data point based on research carried out.  
As stated above, keeping to schedule has worked well, however, this has also caused some stress at times due to being so strict at keeping to the schedule, this has caused stress when other parts of life have been busy, for example, when having very busy weeks at work and also keeping up to schedule with PLePA at the same time.

## Review of project management

### 2.2.1 Overview of chosen SDLC

The lifecycle choice, structured-case lifecycle, has worked very well for PLePA. The main reason for this is because each bit of work could be split into a separate conceptual framework. This has meant the project has followed a chronological order set out, if the project was to fall behind, the later CFs would not get completed. This is ok because during the planning stages, the later CF choices were nice to haves more than necessity’s to PLePA. The disadvantage of this lifecycle is that the later CFs may not even get worked on or the project may be midway through a CF at the deadline.

### 2.2.2 Planning issues

The first, see Appendix 5, and current schedule, see Appendix 6, differ slightly. The first difference was to adjust when some of the work would start for CF2, this was to get TMA02 completed in time for the deadline. Quite a big change was the adjustment of the lengths of CF3, 4 and 5. This was because, initially, CF5 had a very small time frame to be completed, it started with overlapping on CF4 and only lasted two full weeks after that. This was adjusted to have three full weeks after realising how long the previous algorithms took with research, development and gathering results. This also has a knock-on effect on CF6 and CF7 meaning they will be less likely to be completed but that is the benefit of adjusting CFs in a structured-case life cycle.

### 2.2.3 Risk and resource management

The full resource list can be seen in Appendix 7. Each resource will be reviewed here.

* Time has been managed appropriately, as previously stated, and PLePA has kept to schedule where possible, not too many changes to the schedule and no drastic changes.
* There have been no PC issues, still have laptop available and all work is stored in GitHub as well as locally.
* Python has been installed everywhere, no Python issues.
* MySQL has been installed everywhere, all database files are backed up, the database runs on the main desktop.
* All data stored in CSV files and in the database. Backed up to GitHub.
* Research has been carried out throughout the project, there was lots of research done at the beginning of PLePA and each CF stage starts with further research.

The full risk list can be seen in Appendix 8. Each risk will be reviewed here.

* Planning on a house move would have taken up some time, however, due to COVID-19, the house move was cancelled so this is no longer a risk. The house move, if it still happens, will be after the deadline of this project.
* Work travel has also been cancelled due to COVID-19 so this is no longer a risk.
* Desktop and laptop failure, as stated above, there have been no computer related issues. Still a risk which needs to be managed.
* Knowledge gap for new Python modules is being handled by plenty of research being carried at the start of each CF so there is a good understanding of the modules required. Still a risk to be managed.
* Unable to gain knowledge on report writing has been managed by speaking to my partner who writes plenty of reports in her work. I have also had to write more lengthy documents at work which has improved my report writing. Still a risk to be managed.
* Premier league data has been downloaded and used, it is currently stored in the database, no longer a risk.
* Time, currently keeping to the latest schedule, still a risk to be managed.

## Review of personal development

### 2.3.1 Good and bad ways of working and why

I have found that the best way of working for me is at weekends usually. This is because I work full time and working in the evenings after a full working day can be less productive, I expect this is due to the amount of crossover between the project and my full-time job. Both use a lot of database and Python work. This meant I did most of my project work at the weekends.   
I also noticed that it was better to do the work when I was in the mood for working, sometimes forcing myself to do work because I felt I had to would lead to less productivity and more daydreaming. I overcame this by working mainly when I was in the mood, this isn’t always possible if you go a week or two lacking motivation at which point, I would just sit down and do the work.

### 2.3.2 Skills to complete work

The full skills list can be seen in Appendix 9, I will review each skill here.

* There have not been issues with Python coding and lack of the skill has been resolved by doing research which has not been laborious due to the enjoyment of writing Python code.
* The SQL for this project was quite basic compared to what I am used to so this skill has not caused any problems.
* Report writing could be better, I have worked on this, as in 2.2.3, by conversing and asking for help from my partner.
* Researching has been an interesting skill to learn, this could still be better but the more that it is done, the more confident I feel about researching the next part required for the project.

**[5481 words]**

# References

Jason Brownlee (August 2019) *How to Implement Random Forest From Scratch in Python* [Online] Available at <https://machinelearningmastery.com/implement-random-forest-scratch-python/> (Accessed 14th April)

Jason Brownlee (December 2019) *How to Implement The Decision Tree From Scratch in Python* [Online] Available at <https://machinelearningmastery.com/implement-decision-tree-scratch-python/> (Accessed 14th April)

Jake Huneycutt (May 2018) *Implementing a Random Forest Classification Model in Python* [Online] Available at <https://medium.com/@hjhuney/implementing-a-random-forest-classification-model-in-python-583891c99652> (Accessed 19th April)

Jake VanderPlas (n.d) *In-Depth: Decision Trees and Random Forests* [Online] Available at <https://jakevdp.github.io/PythonDataScienceHandbook/05.08-random-forests.html> (Accessed 19th April)

Cory Maklin (July 2019) *K Nearest Neighbor Algorithm In Python* [Online] Available at <https://towardsdatascience.com/k-nearest-neighbor-python-2fccc47d2a55> (Accessed 17th May)

Lars Buitinck et al. (n.d) 1.6 Nearest Neighbors [Online] Available at <https://scikit-learn.org/stable/modules/neighbors.html> (Accessed 17th May)

Jason Brownlee (October 2019) Develop k-Nearest Neighbors in Python from Scratch [Online] Available at <https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/> (Accessed 17th May)

Avinash Navlani (August 2018) KNN Classifciation using Scikit-learn [Online] Available at <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn> (Accessed 17th May)

# Bibliography

Dataflair Team (September 2018) *8 Machine Learning Algorithms in Python* [Online]. Available at <https://data-flair.training/blogs/machine-learning-algorithms-in-python/> (Accessed 16th February 2020)

J.M.Carroll, P.A.Swatman (2000) *Structured Case: A Methodological Framework for Building Theory in Information Systems Research*. Available at <https://pdfs.semanticscholar.org/caff/950b0efc8f23e17e7de171d8836c8dfa1044.pdf>(Accessed 17th February 2020)

The Open University (2014) *Choosing a Lifecyle Model*, *TM470* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1557365>(Accessed 2nd February 2020)

Victor Osetskyi (2017) *SDLC Models Explained: Agile, Waterfall, V-Shaped, Iterative, Spiral* [Online]. Available at <https://medium.com/existek/sdlc-models-explained-agile-waterfall-v-shaped-iterative-spiral-e3f012f390c5>(Accessed 14th February 2020)

The Open University (2019) ‘Data mining I: classification tasks’ *The k-nearest neighbours algorithm, TMA351* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1349974&section=3> (Accessed 24th February 2014)

The Open University (2019) ‘Data mining II: clustering tasks’ *Partitional clustering: the k-means algorithm, TM351* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1349975&section=2.2> (Accessed 24th February 2014)

Football-Data (February 2020) *Data-files: England* [Online]. Available at <http://www.football-data.co.uk/englandm.php> (Accessed 24th February)

FBREF (February 2020) *2019-2020 Premier League Stats* [Online]. Available at <https://fbref.com/en/comps/9/Premier-League-Stats> (Accessed 24th February)

Tuan Doan Nguyen (August 2018) *The Beautiful Game: Predicting the Premier League with a random model* [Online]. Available at <https://towardsdatascience.com/o-jogo-bonito-predicting-the-premier-league-with-a-random-model-1b02fa3a7e5a> (Accessed 24th February)

Nick Campanelli (May 2019) *Betting on the English Premier League* [Online]. Available at <https://towardsdatascience.com/betting-on-the-english-premier-league-making-money-with-machine-learning-fb6938760c64> (Accessed 24th February)

TM470 course team (2012) *‘Legal, Social, Ethical and Professional issues’ in TM470 study material, The Open University, Milton Keynes* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1184585> (Accessed 1st April)

UK Government (2018) *The Data Protection Act* [Online]. Available at <https://www.gov.uk/data-protection> (Accessed 1st April)

UK Government (2015) *Equality Act 2010: guidance* [Online]. Available at <https://www.gov.uk/guidance/equality-act-2010-guidance> (Accessed 1st April)

UK Government (1990) *Computer Misuse Act 1990* [Online]. Available at <http://www.legislation.gov.uk/ukpga/1990/18> (Accessed 1st April)

UK Government (n.d) *How copyright protects your works* [Online]. Available at <https://www.gov.uk/copyright> (Accessed 1st April)

Information Commissioner’s Office (ICO) (n.d) *What is the Freedom of Information Act?* [Online]. Available at <https://ico.org.uk/for-organisations/guide-to-freedom-of-information/what-is-the-foi-act/> (Accessed 1st April)

BCS, The Charted Institute for IT (June 2019) *Code of Conduct for BCS Members* [Online] Available at <https://cdn.bcs.org/bcs-org-media/2211/bcs-code-of-conduct.pdf> (Accessed 1st April)

Will Koehrsen (August 2018) *An Implementation and Explanation of the Random Forest in Python* [Online] Available at <https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76> (Accessed 5th April)

Synced (October 2017) *How Random Forest Algorithm Works in Machine Learning* [Online] Available at <https://medium.com/@Synced/how-random-forest-algorithm-works-in-machine-learning-3c0fe15b6674> (Accessed 5th April)

Saimadhu Polamuri (May 2017) *How the Random Forest Algorithm Works in Machine Learning* [Online] Available at <https://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/> (Accessed 5th April)

Jason Brownlee (August 2019) *How to Implement Random Forest From Scratch in Python* [Online] Available at <https://machinelearningmastery.com/implement-random-forest-scratch-python/> (Accessed 14th April)

Jason Brownlee (December 2019) *How to Implement The Decision Tree From Scratch in Python* [Online] Available at <https://machinelearningmastery.com/implement-decision-tree-scratch-python/> (Accessed 14th April)

Jake Huneycutt (May 2018) *Implementing a Random Forest Classification Model in Python* [Online] Available at <https://medium.com/@hjhuney/implementing-a-random-forest-classification-model-in-python-583891c99652> (Accessed 19th April)

Jake VanderPlas (n.d) *In-Depth: Decision Trees and Random Forests* [Online] Available at <https://jakevdp.github.io/PythonDataScienceHandbook/05.08-random-forests.html> (Accessed 19th April)

Cory Maklin (July 2019) *K Nearest Neighbor Algorithm In Python* [Online] Available at <https://towardsdatascience.com/k-nearest-neighbor-python-2fccc47d2a55> (Accessed 17th May)

Lars Buitinck et al. (n.d) 1.6 Nearest Neighbors [Online] Available at <https://scikit-learn.org/stable/modules/neighbors.html> (Accessed 17th May)

Jason Brownlee (October 2019) Develop k-Nearest Neighbors in Python from Scratch [Online] Available at <https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/> (Accessed 17th May)

Avinash Navlani (August 2018) KNN Classifciation using Scikit-learn [Online] Available at <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn> (Accessed 17th May)

# Appendices

## Appendix 1 – Snippet of project journal

**17th May**

Research on second algorithm, KNN

**24th May**

Begun development on KNN algorithm.  
Added journal entries for 9th, 11th, 14th, 17th and 24th May

**25th May**

More work on KNN algorithm.

**30th May**

More work on KNN algorithm.

**31st May**

Completed KNN algorithm

**6th June**

Collect data on KNN algorithm

**9th June**

Collect more data on KNN algorithm

**12th June**

Review KNN algorithm.  
Added journal entries since 24th May.

## Appendix 2 – LEPSI Review

|  |  |  |  |
| --- | --- | --- | --- |
| Act / Law / Guidance | Purpose | Relevance to my project or n/a and why not applicable | How my project will be affected |
| Data Protection Act 2018 (DPA) | Control how personal information is used by organisations, businesses or the government (UK Government, 2018) | I will not be using any personal information in my project, so this is not applicable for me. | My project will be unaffected. |
| Equality Act 2010 | Protection from discrimination in the workplace and society (UK Government, 2012) | I will not be looking at people in my project so this will not be applicable for me. | My project will be unaffected. |
| Computer Misuse Act 1990 | The act was brought into place to prevent people gaining unauthorised access to computer material, commonly known as hacking. (UK Government, 1990) | My project will likely not involve connection to the internet. It may do, if I reach my stretch goal to connect to a sports website for latest fixtures. | If I reach my stretch goal for connection to the sports website to get the latest fixtures, I will ensure the connection I have is secure and authorised. It likely will be authorised since the website will be publishing it. |
| Copyright | To protect your work by preventing it being copied, redistributed, adapted and put on the internet are some of the example (UK Government, n.d) | I will need to make sure the data that I use for past football results is not protected by copyright and can be used. | I will need to ensure I am able to use the data and not breaking copyright rules. |
| Freedom of Information Act | It gives the general public access to certain information on request from the public authorities. The public authorities are also obliged to publish certain information. (ICO, n.d) | This will not affect my project because I am not working for a public authority and I don’t require data from a public authority. | My project will be unaffected. |
| Protection from harm | Protect participants of studies from any harm, physical or psychological, particular care should be paid to children. (TM470 course team, 2012) | This will not affect my project because I’m not using participants. | My project will be unaffected. |
| Professional codes of practice and ethics | This allows participants to understand the purpose of the study, the researchers must state their intentions. (TM470 course team, 2012) | This will not affect my project because I’m not using participants. | My project will be unaffected. |
| BCS Code of Conduct | To set out standards across the board for all members of the BCS. An example is having respect for public health, privacy, security and wellbeing of others and the environment. (BCS, June 2019) | This will not affect me directly because I’m not a member of the BCS, however, I should try to comply with the standards set out by the BCS such as honesty with my skillset and acknowledgment to any borrowed source code used in my project. | I must ensure that borrowed code is acknowledged and I am not overexaggerating my skillset. |

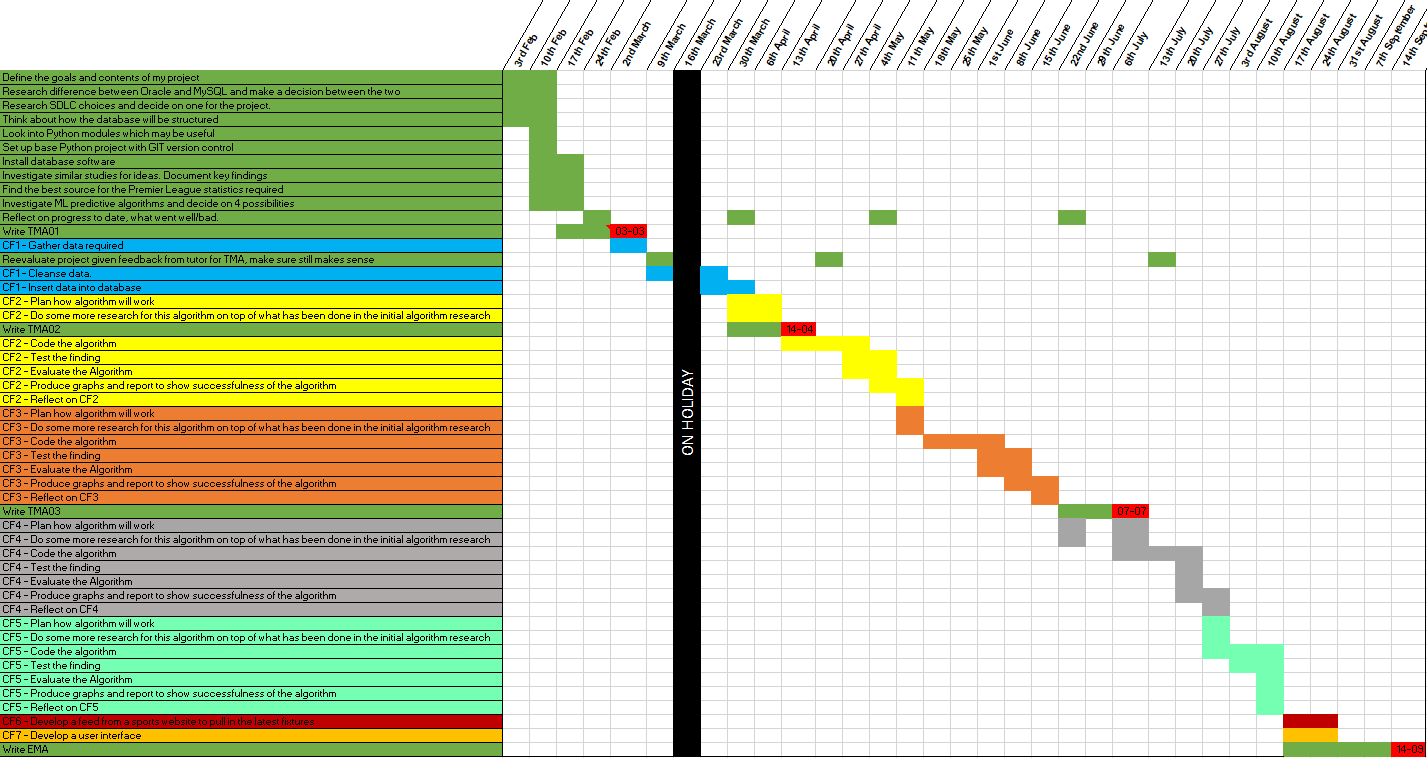
## Appendix 3 – Table creation script

CREATE TABLE stadium (  
 stadium\_id\_pk *INT* AUTO\_INCREMENT PRIMARY KEY,  
 stadium\_name *VARCHAR*(255) NOT NULL,  
 x\_coord *DECIMAL*(13,8) NOT NULL,  
 y\_coord *DECIMAL*(13,8) NOT NULL  
);  
  
CREATE TABLE team (  
 team\_id\_pk *INT* AUTO\_INCREMENT PRIMARY KEY,  
 name\_pk *VARCHAR*(255) NOT NULL,  
 stadium\_id\_fk *INT*,  
 CONSTRAINT fk\_stadium  
 FOREIGN KEY (stadium\_id\_fk)  
 REFERENCES stadium(stadium\_id\_pk)  
);  
  
CREATE TABLE season\_overview (  
 team\_id\_pk\_fk *INT*,  
 season\_pk *INT*,  
 position *INT*,  
 goals\_for *INT*,  
 goals\_against *INT*,  
 wins *INT*,  
 draws *INT*,  
 losses *INT*,  
 PRIMARY KEY (team\_id\_pk\_fk, season\_pk),  
 CONSTRAINT fk\_team\_id  
 FOREIGN KEY (team\_id\_pk\_fk)  
 REFERENCES team(team\_id\_pk)  
);  
  
CREATE TABLE game (  
 game\_pk *INT* AUTO\_INCREMENT,  
 home\_team\_id\_pk\_fk *INT*,  
 away\_team\_id\_pk\_fk *INT*,  
 season\_pk *INT*,  
 home\_team\_score *INT*,  
 away\_team\_score *INT*,  
 PRIMARY KEY (game\_pk, home\_team\_id\_pk\_fk, away\_team\_id\_pk\_fk, season\_pk),  
 CONSTRAINT fk\_home\_team\_id  
 FOREIGN KEY (home\_team\_id\_pk\_fk)  
 REFERENCES team(team\_id\_pk),  
 CONSTRAINT fk\_away\_team\_id  
 FOREIGN KEY (away\_team\_id\_pk\_fk)  
 REFERENCES team(team\_id\_pk)  
);

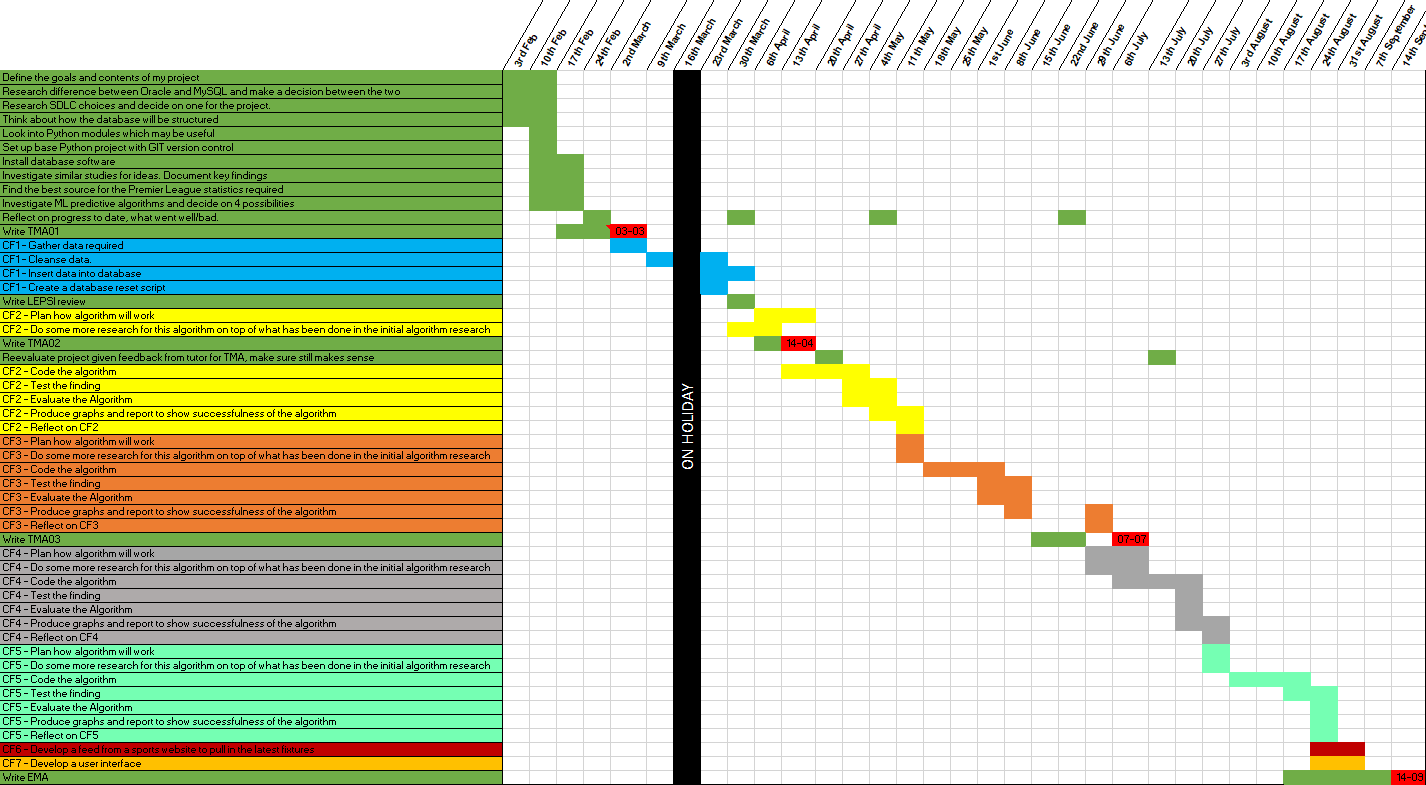
## Appendix 4 - Main function for database insert

def main():  
 plepa\_db = mysql.connector.connect(  
 host='localhost',  
 user='root',  
 passwd='PLePApw',  
 database='plepa'  
 )  
  
 db\_cursor = plepa\_db.db\_cursor()  
  
 insert\_data('stadium\_team', STADIUM\_DATA, db\_cursor, plepa\_db)  
 insert\_data('game', GAME\_DATA, db\_cursor, plepa\_db)  
 insert\_data('season\_overview', SEASON\_OVERVIEW\_DATA, db\_cursor, plepa\_db)  
  
 plepa\_db.close()

## Appendix 5 – First draft of schedule

****

## Appendix 6 – Latest draft of schedule



## Appendix 7 – Resource list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resource | Why needed | When needed | Problems if not available | How to ensure availability |
| Time | Complete any task for the project. | Throughout project. | The less time available, the fewer tasks can be completed. | Cannot ensure time but can try to get ahead on tasks in case of any complications. |
| PC | To complete all tasks, requires pc availability. | Throughout project. | Whilst no PC is available, most tasks will be unable to be achieved. | I have a desktop and laptop. As a last resort, I can also use my work laptop. |
| Python | Write the code for the algorithms. | During CF2-5. | The code for the algorithms will not be able to be written. | It is installed on all PCs available. |
| MySQL | Store the data used for algorithms. | Set up in CF1 but will be required from CF1 – CF7. | The algorithms will have no database to read from. | It is installed on all PCs available. |
| Premier League data | The project is using data from the Premier League for the algorithms. | Pre CF1, it will then be stored in the MySQL database. | No data to be used for the algorithms. | Once gathered and stored in the database, store a copy of the database. |
| Information from previous studies | To understand what went well and not so well for similar projects. | Studied prior to any real task being completed, used throughout. | No knowledge from other teams’ experiences so PLePA may make similar mistakes as other studies which were not investigated. | Look on the internet and in the library early in the project. |

## Appendix 8 – Risk list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk involved | Priority – low to high | Overcome/manage risk | What to do if not managed | Review of risk |
| Planning on a house move. | Medium | Get ahead on project when possible, manage time during move. | Make up the time after the house move is complete. | Accept – try to manage time accordingly around busy times for house move. |
| Work travel, 2-4 days a month. | Low | Manage workload accordingly around work travel to get ahead. | Try to find a bit of extra time following the travel to catch up on project. | Accept – try to get ahead before work travel and catch up if needed afterwards. |
| Desktop and laptop failure | Low | Data stored at GitHub and locally on both machines. | Library work or purchase new laptop/desktop to replace. | Avoid – Unlikely that both desktop and laptop will fail. |
| Knowledge gap on new Python modules | High | Research the modules required, also have a Python expert at my work who I can consult with. | Lower the detail of the algorithms used. | Accept – start researching required modules early and learn how to use. |
| Unable to gain knowledge on report writing | High | Lots of preparation and research ahead of the EMA, the TMAs are good practice too. | A poor representation of work done, won’t be reflective of the work put in in the project. | Avoid – make sure to research and ask for advice from contacts who are good at report writing. |
| Unable to obtain Premier League data | High | Ensure the data is available and download onto PC before lost. | No data available for the project, very difficult to continue | Avoid – it would be disastrous to not obtain the data required. |
| Time | Medium | Manage time correctly, try to productive when working and keep up with the schedule. | Project may fall behind, and milestones may not be met. The project could be unfinished. | Accept – manage time appropriately, try not to fall too far behind on schedule. If behind on schedule, try to catch up. |

## Appendix 9 – Skill list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skill | Why needed | When needed | Problems if not available | How to ensure availability |
| Python coding | Write the code for the algorithms. | During CF2-5. | The code for the algorithms will not be able to be written. | Research unused additional Python modules which are required. |
| SQL coding | Write the code for the database. | During CF1. | Unable to create the database structure and insert the data. | Used in everyday work, fluent in SQL. |
| Time management | Ensure milestones are met. | Throughout project. | Milestones may be missed; project will be incomplete. | Research good time management ideas and seek advice from people with good time management. |
| Report writing | To write TMA1-3 and EMA. | For all assignments but mainly the EMA. | The quality of writing for TMAs and EMAs will be lacking. | Research and check previous module’s advice on report writing and read course booklet on the subject |
| Researching | For several tasks to investigate best approach | Throughout project. | Project will suffer due to decisions being made without right information. | Look into ideas for researching, the best approaches to research and read course booklet on the subject |